

AMENDMENT OF THE CLAIMS

The listing of claims below replaces all prior versions, and listings, of claims:

1 1. (Cancelled)

1 2. (Previously Presented) The method of claim 7, wherein receiving a data
2 transaction request comprises receiving a request for loading data into the database.

1 3. (Previously Presented) The method of claim 7, wherein receiving a data
2 transaction request comprises receiving a request to perform a data transformation
3 operation upon the data in the database.

1 4. (Original) The method of claim 3, wherein receiving a request to perform
2 the data transformation operation comprises receiving a request to perform one of a data
3 selection operation, a data validation operation, a data cleansing operation, and a data
4 query operation.

1 5. (Cancelled)

1 6. (Previously Presented) The method of claim 7, comprising executing a
2 first parallel task in a first number of phases and a second parallel task in a second,
3 different number of phases.

1 7. (Previously Presented) A method of performing parallel data operations
2 upon data in a database, comprising:

3 receiving a data transaction request in a client system;
4 executing a plurality of multi-phase parallel tasks in response to the
5 request to perform the data operations upon the data in the database, wherein executing
6 the multi-phase parallel tasks comprises executing each of the parallel tasks in plural
7 phases; and

8 each parallel task providing a code to indicate if the task is to be re-
9 invoked in the next phase.

1 8. (Original) The method of claim 7, wherein providing the code comprises
2 providing the code to a task coordinator.

1 9. (Original) The method of claim 8, wherein the code comprises a first code
2 to indicate that the task coordinator is to invoke a component in the next phase.

1 10. (Original) The method of claim 8, wherein the code comprises a second
2 code to indicate that the task is not to invoke a component in the next phase.

1 11. (Currently Amended) A method of performing parallel data operations
2 upon data in a database, comprising:
3 receiving a data transaction request in a client system;
4 executing a plurality of multi-phase parallel tasks in response to the
5 request to perform the data operations upon the data in the database;
6 analyzing the transaction request;
7 creating a task plan in response to the transaction request;
8 implementing the task plan in a multi-phase organization, wherein the
9 plurality of multi-phase parallel tasks are executed to implement the task plan;
10 ~~executing a plurality of tasks in parallel to implement the task plan;~~
11 determining whether an additional phase is required to execute the tasks
12 based on codes returned by the tasks; and
13 scheduling an additional phase in response to the determination that an
14 additional phase is required.

1 12. (Original) The method of claim 11, wherein implementing the task plan
2 comprises creating a job script.

1 13. (Currently Amended) The method of claim 11, wherein implementing the
2 task plan comprises:
3 translating the task plan;
4 selecting a plurality of software components corresponding to the parallel
5 tasks to implement the translated task plan;
6 assigning a plurality of processes corresponding to the software
7 components; and
8 creating a communications channel to allow for communications between
9 the processes.

1 14. (Previously Presented) The method of claim 13, wherein selecting the
2 plurality of software components to implement the translated task plan comprises
3 selecting the plurality of software components to perform at least one of a data extraction
4 operation, a data transformation operation, and a data loading operation.

1 15. (Previously Presented) An apparatus, comprising:
2 a user interface;
3 a processor coupled with the user interface, wherein the processor receives
4 a data transaction request from the user interface; and
5 a controller coupled with the processor, wherein the controller performs a
6 plurality of tasks in parallel based upon instructions received from the processor, each
7 task performed in a plurality of phases,
8 each task to provide a code to indicate whether the task is to be re-invoked
9 in a next phase.

1 16. (Original) The apparatus of claim 15, wherein the processor generates a
2 task plan in response to the data transaction request.

1 17. (Original) The apparatus of claim 16, wherein the controller comprises a
2 task coordinator to execute the task plan.

1 18. (Original) The apparatus of claim 16, wherein the controller further
2 comprises a plurality of components to implement the task plan in parallel.

1 19. (Original) The apparatus of claim 18, further comprising a
2 communications interface for enabling communications between the components.

1 20. (Original) The apparatus of claim 18, wherein the controller further
2 comprises a storage unit for storing methods and functions to execute the task plan.

1 21. (Currently Amended) The apparatus of claim 15, ~~wherein the controller is~~
2 ~~coupled with the processor,~~ wherein the controller performs a number of tasks in parallel
3 based upon instructions received from the processor, each task performed in a plurality of
4 phases further comprises the controller performing the tasks in a sequence of multiple
5 process steps.

1 22. (Previously Presented) A system, comprising:
2 a database system;
3 a network; and
4 a client system separate from the database system and coupled to the
5 database system over the network, the client system to establish plural sessions with the
6 database system to implement a plurality of data operations upon the database system in
7 parallel.

1 23. (Cancelled)

1 24. (Previously Presented) The system of claim 22, wherein the database
2 system is a parallel database system.

1 25. (Previously Presented) The system of claim 22, wherein the client system
2 comprises:
3 a processor to receive a data transaction request;
4 a plurality of operators to perform parallel data operations in response to
5 the data transaction request;
6 an operator interface coupled to the operators, wherein the operator
7 interface allows communications between the operators.

8
1 26.-27. (Cancelled)

1 28. (Previously Presented) An article comprising at least one storage medium
2 containing instructions that when executed cause a client system to:

3 receive a data transaction request;

4 establish plural sessions with a database system over the network
5 connection in response to the request; and

6 execute a plurality of parallel tasks in the plural sessions to perform data
7 operations upon the data in the database system over a network connection, wherein the
8 client system is separate from the database system.

1 29. (Previously Presented) The article of claim 28, wherein the instructions
2 when executed cause the client system to execute each of the parallel tasks in plural
3 phases.

1 30. (Previously Presented) The article of claim 29, wherein the instructions
2 when executed cause the client system to execute a first parallel task in a first number of
3 phases and a second parallel task in a second, different number of phases.

1 31. (Original) The article of claim 29, wherein the instructions when executed
2 cause each parallel task to provide a code to indicate if the task is to be re-invoked in the
3 next phase.

1 32. (Original) The article of claim 31, wherein the instructions when executed
2 cause the parallel task to provide the code to a task coordinator.

1 33. (Original) The article of claim 32, wherein the code comprises a first code
2 to indicate that the task coordinator is to invoke a component in the next phase.

1 34. (Original) The article of claim 32, wherein the code comprises a second
2 code to indicate that the task is not to invoke the component in the next phase.

1 35.-36. (Cancelled)

1 37. (Previously Presented) The method of claim 38, comprising executing a
2 first parallel task in a first number of phases and a second parallel task in a second,
3 different number of phases.

1 38. (Previously Presented) A method of performing parallel data operations
2 upon data in a database, comprising:

3 receiving a data transaction request; and
4 executing a plurality of synchronized multi-phase parallel tasks in
5 response to the request to perform the data operations upon the data in the database,
6 wherein executing the multi-phase parallel tasks comprises executing each
7 of the parallel tasks in phases; and
8 each parallel task providing a code to indicate if the task is to be re-
9 invoked in the next phase.

10
1 39. (Original) The method of claim 38, wherein providing the code comprises
2 providing the code to a task coordinator.

1 40. (Original) The method of claim 39, wherein the code comprises a first
2 code to indicate that the task coordinator is to invoke a component in the next phase.

1 41. (Original) The method of claim 39, wherein the code comprises a second
2 code to indicate that the task is not to invoke a component in the next phase.

1 42. (Previously Presented) The method of claim 39, wherein the code
2 synchronizes the operation of one or more components.

1 43. (Previously Presented) The method of claim 7, wherein executing the
2 plurality of multi-phase parallel tasks comprises:
3 executing at least first and second software components in parallel;
4 each of the first and second software components performing one or more
5 operations in a first phase;
6 waiting for a message comprising the code from each of the first and
7 second software components prior to proceeding to a second phase; and
8 each of the first and second software components performing one or more
9 operations in the second phase.

1 44. (Previously Presented) The method of claim 43, further comprising:
2 waiting for another message from each of the first and second software
3 components prior to proceeding to a third phase;
4 the first software component performing one or more operations in the
5 third phase; and
6 the second software component being idle in the third phase.

1 45. (Previously Presented) The method of claim 44, further comprising:
2 receiving a first message from the first software component indicating that
3 the first software component is to be re-invoked in the third phase; and
4 receiving a second message from the second software component
5 indicating that the second component is not to be re-invoked in the third phase.

1 46. (Previously Presented) The apparatus of claim 15, wherein the controller
2 comprises at least first and second software components executable in parallel to perform
3 the plurality of tasks;
4 wherein each of the first and second software components is executable to
5 perform one or more operations in a first phase;
6 the controller to wait for a message comprising the code from each of the
7 first and second software components prior to proceeding to a second phase; and

8 wherein each of the first and second software components is executable to
9 perform one or more operations in the second phase.

1 47. (Previously Presented) The apparatus of claim 46, wherein the controller
2 is adapted to further wait for another message from each of the first and second software
3 components prior to proceeding to a third phase;

4 wherein the first software component is executable to perform one or more
5 operations in the third phase, and the second software component is idle in the third
6 phase.

1 48. (Previously Presented) The apparatus of claim 47, wherein the controller
2 is adapted to further:

3 receive a first message from the first software component indicating that
4 the first software component is to be re-invoked in the third phase; and

5 receive a second message from the second software component indicating
6 that the second component is not to be re-invoked in the third phase.

1 49. (Previously Presented) The system of claim 22, wherein the client system
2 is adapted to execute plural tasks in parallel, each of the plural tasks executable in plural
3 phases.

1 50. (Previously Presented) The article of claim 29, wherein executing each of
2 the parallel tasks in plural phases comprises:

3 executing at least first and second software components in parallel;

4 each of the first and second software components performing one or more
5 operations in a first phase;

6 waiting for a message from each of the first and second software
7 components prior to proceeding to a second phase; and

8 each of the first and second software components performing one or more
9 operations in the second phase.

1 51. (Previously Presented) The article of claim 50, wherein the instructions
2 when executed cause the client system to further:

3 wait for another message from each of the first and second software
4 components prior to proceeding to a third phase;

5 cause the first software component to perform one or more operations in
6 the third phase; and

7 cause the second software component to be idle in the third phase.

1 52. (Previously Presented) The article of claim 51, wherein the instructions
2 when executed cause the client system to further:

3 receive a first message from the first software component indicating that
4 the first software component is to be re-invoked in the third phase; and

5 receive a second message from the second software component indicating
6 that the second component is not to be re-invoked in the third phase.

1 53. (New) The method of claim 7, wherein executing the plurality of multi-
2 phase parallel tasks includes executing a plurality of checkpoint tasks in parallel, each
3 checkpoint task having multiple phases, and each checkpoint task to write data to storage
4 to provide an indication of a current execution point.

1 54. (New) The method of claim 7, wherein executing the plurality of multi-
2 phase parallel tasks includes executing the plurality of multi-phase tasks in parallel.

1 55. (New) The apparatus of claim 15, wherein the plurality of tasks include a
2 plurality of checkpoint tasks that are executed in parallel, each checkpoint task having
3 multiple phases, and each checkpoint task to write data to storage to provide an indication
4 of a current execution point.